PyCIRCLean: a versatile Python framework to check and/or sanitize files

CIRCL
Computer Incident
Response Center
Luxembourg

TLP:WHITE

info@circl.lu

March 7, 2016

Overview

- Aims to be used in dedicated security applications to sanitize documents from hostile to trusted environments.
- Generic way to handle large colections of files
- Generate audit logs
- Comes with many helpers

Implementation

- Copies files from a directory (source) to an other one (destination)
- Computes hashes (sha1) of all the files in the source
- Creates a directory tree on the destination directory
- Gets the mime type of each file

Existing modules

- bin/filecheck.py: Search for active content in the source documents
- bin/generic.py: Converts documents if possible
- bin/specific.py: Only copy a specific extension if the mimetype matches
- bin/pier9.py: Only copy specific extensions (3D softwares)

File Check

- Discard known extensions with active content
- Verifies if the extension corresponds to the mimetype (polyglot files)
- Force extension on suposedly text files
- Discards windows executables
- Discard Office (Libreoffice and Windows Office) document with active content
- Discard PDFs with active content
- Unpack archives and process content
- Extract metadata from images

File Check

- Plus
  - (almost) Pure python
  - Reliable
  - Fast
- Minus
  - Does not block a 0 day in a non-active content
  - Medium level of false positive (non-malicious active content)

Generic

- Verifies if the extension corresponds to the mimetype (polyglot files)
- Converts to PDF and then to HTML all documents supported by libreoffice
- Converts to HTML all PDF files
- Discards windows executables
- Unpack archives and process content

Generic

- Plus
  - Very hard to have anything malicious in the output of the converted documents
- Minus
  - Slow
  - Opens the documents to convert (may run malicious code)
  - Many external dependencies
  - Unreliable: fails on 20% of the documents

Specific and Pier9

- Dedicated to a very specific use
- Whitelist on extension and/or MimeType
- Plus
  - Pure python
  - Very fast
  - Most secure
- Minus
  - Only works in a specific case
  - Many false positive

Implement your own module - FileBase

- The default conctructors gets the mime type of the file and initialize the log of the file
- Surcharge the constructor accordingly to your needs
- Has helpers to get and set information on the file being processed
- Can force the extension of the file when copied
- All thoses functions have to be used in order to handle the files accordingly to your requirements

Implement your own module - KittenGroomerBase

- The default constructor cleans the destination directory, starts the general logging and logs the content of the source directory
- Has helpers to handle safely the file management
- Writes the logs files